

(Re-)Allokationsstrategien für Aufträge mit alternativen Ausführungsmodi – Simulationsexperimente zur Transportlogistik

Jörn Schönberger

Veröffentlicht in:
Multikonferenz Wirtschaftsinformatik 2012
Tagungsband der MKWI 2012
Hrsg.: Dirk Christian Mattfeld; Susanne Robra-Bissantz



Braunschweig: Institut für Wirtschaftsinformatik, 2012

(Re-)Allokationsstrategien für Aufträge mit alternativen Ausführungsmodi – Simulationsexperimente zur Transportlogistik

Jörn Schönberger

Universität Bremen, Lehrstuhl für Logistik, 28359 Bremen, E-Mail: jsb@uni-bremen.de

Abstract

Volatile Entscheidungskonstellationen werden typischerweise als eine Sequenz miteinander verknüpfter statischer Entscheidungsinstanzen (d.h. als Online-Entscheidungsproblem) repräsentiert. Diese Instanzen werden sukzessive bearbeitet. Eine wesentliche Herausforderung, die bei der Definition von Entscheidungsunterstützungssystemen für Online-Entscheidungsprobleme bewältigt werden muss, besteht in der Bewältigung der logischen Abhängigkeiten zwischen den nacheinander bearbeiteten Instanzen. In diesem Artikel untersuchen wir ein Online-Entscheidungsproblem aus der Transportlogistik. Es wird untersucht, ob einmal getroffene Allokationsentscheidungen in späteren Instanzen modifiziert oder beibehalten werden sollten. Verschiedene Re-Allokationsstrategien werden vorgeschlagen und in computerbasierten Simulationsexperimenten evaluiert.

1 Einleitung

Die Analyse und Klassifikation eintreffender Kundennachfragen sind wichtige Schritte in der Planung und Modifikation von Transportprozessen. Aufträge, für die einem Kunden die Erfüllungszusage gegeben wurde, können nachträglich nicht mehr zurückgegeben werden. Ein akzeptierter Kundenauftrag „konsumiert“ darüber hinaus die nur limitiert vorhandenen Transportkapazitäten. Im Rahmen einer Auftragsannahmeentscheidung ist durch einen Transport-Dienstleister zu entscheiden, ob die verfügbaren Restkapazitäten für die aktuelle Kundenanfrage allokiert werden oder ob sie für zukünftige noch erwartete Auftragsanfragen freigehalten werden, für die ein höherer Erlös erwartet wird.

Auftragsannahmeentscheidungen stellen dynamische Entscheidungsprobleme dar, denn Kundenanfragen werden sukzessive gestellt und die Annahmeentscheidungen müssen sequentiell und unumkehrbar getroffen werden. Falls es einem Transport-Dienstleister möglich ist, die Allokation von Ressourcen für einen Auftrag nach dessen unumkehrbarer Annahme noch zu ändern, so wird das nachgefragte Produkt als *flexibel* bezeichnet [1]. Für eine Nachfrage nach einem flexiblen Produkt ist ein Ausführungsmodus aus zwei oder mehr Modi auszuwählen.

Aus der Sicht eines Prozessplaners eröffnen flexible Produkte zusätzliche Entscheidungsspielräume auch für bereits akzeptierte Kundenanfragen. Daher wird flexiblen Produkten das Potential zur Steigerung der Effizienz des betrachteten Wertschöpfungssystems zugesprochen [3]. Allerdings wurde in Simulationsexperimenten auch beobachtet, dass eine Veränderung der Fahrzeug-Zuordnung von angenommenen Aufträgen (Re-Allokation) insgesamt sogar zu einer Reduzierung von Erlösen und damit zu Profitschmälerungen führen kann [6]. In diesem Artikel wollen wir die Gründe für dieses unerwünschte Systemverhalten hinterfragen und stellen in diesem Zusammenhang die folgende (allgemeine) Forschungsfrage: *Sollten einmal getroffene Entscheidungen über die Zuordnung von akzeptierten Aufträgen zu Fahrzeugen beibehalten (geschützt) werden oder sollten diese Zuordnungsentscheidungen im Rahmen einer rollierenden Planung verändert werden dürfen?*

Die vorgenannten Beobachtungen motivieren die folgenden Forschungshypothesen, die im Rahmen dieses Artikel für ein dynamisches Auftragsannahme-Szenario aus dem Straßengüterverkehr überprüft werden sollen: *Falls die Auftragsannahme aufgrund von Profitabilitätsüberlegungen erfolgt, so ist eine Veränderung einer initial getroffenen Ressourcenallokation nicht vorteilhaft (H_1). Falls solche Allokations-Modifikationen nicht komplett verhindert werden können, so sollte die Anzahl dieser Modifikationen möglichst klein sein (H_2).*

Zunächst wird das untersuchte dynamische Auftragsannahme-Szenario aus der Transportlogistik vorgestellt (Abschnitt 2). Ein Kapazitätssteuerungs-System, dass die Profitabilität eintreffender Auftragsanfragen überprüft wird in Abschnitt 3 skizziert. Re-Allokationsstrategien für flexible Produkte werden in Abschnitt 4 vorgestellt. Simulationsexperimente und Ergebnisse sind Gegenstand von Abschnitt 5.

2 Dynamische Entscheidung für die Annahme von Aufträgen

Zunächst wird der Begriff „Kapazitätssteuerung“ erläutert (Unterabschnitt 2.1). Anschließend erfolgt die Skizzierung des untersuchten Entscheidungsproblems (Unterabschnitt 2.2). Die Vorstellung des Konzepts der parameterbasierten Auftragsannahme-Entscheidung wird in Unterabschnitt 2.3 durchgeführt. Dieser Abschnitt schließt mit der Beschreibung der Bestimmung und der Aktualisierung der benötigten Parameter für die Auftragsannahme (Unterabschnitt 2.4).

2.1 Kapazitätssteuerung

Unter dem Begriff Kapazitätssteuerung werden Entscheidungsunterstützungswerkzeuge für Auftragsannahme-Entscheidungen zusammengefasst, die auf mathematischen Modellen basieren [8]. Zwei generelle Ansätze werden in der Literatur beschrieben und untersucht. Einerseits werden Kontingente festgelegt, die die maximal erlaubte Anzahl von Anfragen für ein Produkt bestimmen [4]. Eine Auftragsanfrage wird akzeptiert, solange das festgelegte Kontingent noch nicht ausgeschöpft ist. Andererseits werden Opportunitätskosten („Bid-Prices“) für die Nutzung von Ressourcen bestimmt, die zur Profitabilitätsprüfung einer Auftragsanfrage herangezogen werden [2]; eine Anfrage wird akzeptiert, falls die mit ihr verbundenen Erträge die berechneten Opportunitätskosten übersteigen und falls noch ausreichend Kapazität für die Auftragsausführung verfügbar ist. Jeder Auftrag wird somit

einer individuellen Profitabilitätsprüfung unterzogen. Dies ist bei der Kontingent-basierten Kapazitätssteuerung nicht der Fall. In diesem Artikel wird eine Bid-Price-basierte Kapazitätssteuerung vorgeschlagen und untersucht.

2.2 Problembeschreibung

Nachfolgend wird eine Kurzdarstellung des untersuchten dynamischen Auftrags-Aannahmeentscheidungs-szenarios gegeben, das im Rahmen dieser Arbeit untersucht wird [6]. Ein Frachtführer (der Transport-Dienstleister) muss sukzessive über die Annahme von Auftragsanfragen von Kunden für Frachttransporte entscheiden. Jeder dieser Aufträge umfasst die Nachfrage nach einem Transport einer unteilbaren Gütermenge durch ein Netzwerk $N:=(V,A)$, das durch verschiedene in der Menge V zusammengefasst Aufladepunkte („Pickup Points“) und Entladepunkte („Delivery Points“) sowie einer Menge A von gerichteten Transportverbindungen (Pfeilen) zwischen ausgewählten Punktpaaren aus V gebildet wird.

Das Netzwerk N wird durch vorab festgelegte Transportservices, die in N ablaufen, definiert. Grundlage der Definition dieser Services sind langfristige Verträge mit verschiedenen Kunden, die regelmäßig einen Besuch durch Fahrzeuge des betrachteten Frachtführers erfordern. Dadurch werden langfristig vereinbarte sog. reguläre Transportleistungen erbracht. Die betrachteten Services werden in der Menge S gesammelt. Jeder Service $s \in S$ wird durch ein Fahrzeug mit der Gesamtladekapazität CAP_s realisiert.

Eine Route ist eine Sequenz von Transportabschnitten und ein Transportabschnitt beinhaltet keine Zwischenstopps für Be-, Ent-, oder Umladeaktivitäten. Jeder Transportabschnitt entspricht genau einem Pfeil $a \in A$. Falls mehrere Fahrzeuge den gleichen Transportabschnitt bedienen, so werden dafür parallele Pfeile in A vorgehalten. Ein Transportabschnitt $a \in A$ wird als **Ressource** interpretiert und $C(a)$ bezeichnet die Kapazität der Ressource, die durch $C(a):=CAP_{s(a)}$ initialisiert wird, wobei $s(a)$ den Service aus S bezeichnet, der entlang a fährt.

Ein **Produkt** p im Netzwerk N ist ein geordnetes Paar zweier Knoten $p:=(u,v) \in V \times V$, wobei mindestens ein Service $s \in S$ Güter am Knoten u aufnimmt, ohne Umladung diese Güter zum Knoten v transportiert und dort vollständig ablädt. Alle Produkte werden in der Menge P zusammengefasst. Der binäre Parameter $SPR(s,p,a)$ ist genau dann 1 wenn Service $s \in S$ das Produkt $p \in P$ bedient und dabei die Ressource $a \in A$ nutzt.

Ein **Auftrag** r drückt die unteilbare Nachfrage nach einem Produkt $p:=P(r) \in P$ aus. Der Frachtführer erwartet einen Erlös $REV(P(r))$ durch die Erfüllung von r .

2.3 Parameter-basierte Auftragsannahme-Entscheidungen

Für jeden sog. *Spot-Markt-Auftrag*, der nicht im Rahmen eines langfristigen Vertrags erfüllt werden muss, kann der Frachtführer frei über die Annahme entscheiden. Er muss dies jedoch sofort nach Übermittlung der entsprechenden Kunden-Anfrage tun. Das Ziel des Frachtführers ist es, aus allen ankommenden Anfragen diejenigen herauszufiltern, die nach unumkehrbarer Annahme die (knappen) Ressourcen am profitabelsten nutzen.

Zu einem Zeitpunkt innerhalb der Buchungs-(bzw. Verkaufs-)Phase erhält der Frachtführer einen oder mehrere Auftragsanfragen aus dem Spot-Markt. Deren Ankunft initiiert einen sog. Request-Cycle (Bild 1). Der Frachtführer muss nun sofort (d.h. bevor weitere Aufträge

bekannt werden) entscheiden, ob ein soeben angekommener Auftragsvorschlag angenommen oder abgelehnt wird.

Zwar kann der Frachtführer eine einmal angenommene Nachfrage r nach einem flexiblen Produkt $P(r)$ nicht mehr zurückgeben, aber er kann für die Ausführung von r einen Servicewechsel durchführen, falls ein anderes Fahrzeug über ausreichend Kapazität verfügt.

Bei ausreichenden Kapazitäten entscheidet sich der Frachtführer zum Zeitpunkt t^k für die Akzeptanz einer Auftragsanfrage r , dann und nur dann, falls die resultierenden Opportunitätskosten (der „Bid-Price“ des Auftrags r) $BP^k(r)$ geringer sind als der Erlös $REV(P(r))$, der durch die Annahme von r erzielt wird, d.h. falls $REV(P(r)) \geq BP^k(r)$. Das sog. **Request Acceptance Tool** (Bild 1) wendet diese Akzeptanzregel an.

Nach erfolgter Annahmeentscheidung beauftragt das Request Acceptance Tool die **Request-to-Service-Assignment**-Komponente des Kapazitätssteuerungssystems mit der (vorläufigen) Auswahl eines geeigneten Services (Fahrzeugs). Die Assignment-Komponente allokiert (d.h. reserviert) für alle bisher akzeptierten Aufträge Ressourcenanteile auf jeweils passenden Fahrzeugen und weist ggf. Aufträge für flexible Produkte einem anderen Fahrzeug zu. Abschließend meldet die Assignment-Komponente die verbleibenden und disponierbaren Kapazitäten an das Request Acceptance Tool zurück.

Die Akzeptanz- und Ablehnungsentscheidungen werden abschließend an die Kunden in der (volatilen) **Prozess-Umgebung** zurückgemeldet. Sobald ein oder mehrere weitere Auftragsanfragen eintreffen, wird der nächste Request Cycle initiiert.

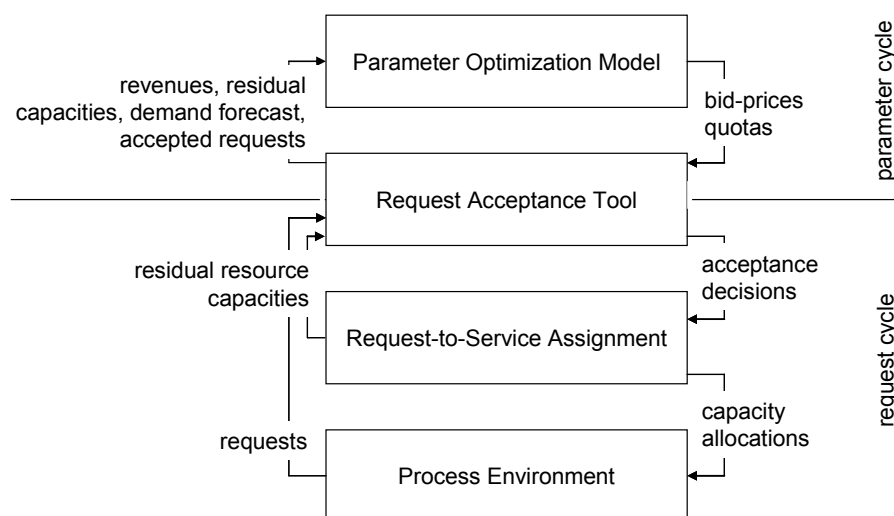


Bild 1: Kapazitätssteuerung mit Parameter-basierter Auftragsannahme

2.4 Modell-basierte Aktualisierung der Annahme-Parameter

Die Bid-Prices bewerten die aktuell noch verfügbaren Ressourcen. Nach der Annahme eines Auftrags verringern sich die Restkapazitäten und müssen daher aktualisiert werden. Da auch nach erfolgter Ablehnung eines Auftrags die zukünftig zu erwartenden Aufträge weniger werden, muss überprüft werden, ob es zukünftig sinnvoller ist, die Ressourcennutzung weiterhin hinauszuzögern (in der Hoffnung, profitablere als die aktuell bekannten Auftragsanfragen zu erhalten). Zunächst muss der Frachtführer die ab dem Zeitpunkt t^k noch erwartete Nachfrage („Demand-to-Come“) $DTC^k(p)$ für alle Produkte $p \in P$ bestimmen und die

Anzahl $Y^k(p)$ der bereits für das Produkt p angenommenen Aufträge bestimmen. Für die Aktualisierung der Bid-Prices wird durch das Request Acceptance Tool ein **Parameter Cycle** initiiert (Bild 1). Unter Verwendung der vorab beschriebenen Daten können die aktualisierten Bid-Prices nun durch die Lösung des nachfolgend beschriebenen *Deterministic Linear Program* (DLP) approximiert werden. Das DLP ermittelt eine Ressourcen-Allokation, mit der unter Beachtung der noch erwarteten Auftragsanfragen und der knappen Ressourcen die insgesamt erwarteten Erlöse maximiert werden. Aus einer optimalen zulässigen Lösung des DLP können Approximationen der neuen Bid-Prices abgeleitet werden. Daher wird die zum Zeitpunkt t^k zu lösende Instanz M^k des DLP als *Parameter-Optimierungs-Modell* (PO-Modell) zum Zeitpunkt t^k bezeichnet. Falls zur Lösung von M^k ein Simplex-Verfahren verwendet wird, so können aus einem optimalen Simplextableau die sog. Schattenpreise $sp^k(a)$ für alle Ressourcen $a \in A$ abgelesen werden [9]. Diese stellen Opportunitätskosten für die Nutzung je einer Kapazitätseinheit dar und können somit als Bid-Price $BP^k(a)$ für die Ressource $a \in A$ verwendet werden.

Das Online-Entscheidungsmodell $M = (M^1, M^2, \dots, M^{N^{replan}})$ [7] stellt somit ein dynamisches Bid-Price-Aktualisierungsproblem dar. Ein Parameter Cycle wird unabhängig von der Ankunft neuer Aufträge angestoßen. Dafür werden N^{replan} Aktualisierungszeitpunkte für die Bid-Prices gleichmäßig über die Buchungsperiode, in der Anfragen akzeptiert werden, verteilt. Sobald ein solcher Aktualisierungszeitpunkt erreicht ist, wird der Parameter Cycle initiiert und damit einerseits die gespeicherten Bid-Prices der Ressourcen aktualisiert. Daneben werden auch service-spezifische Kontingente $Q(p,s)$ ($p \in P, s \in S$) für die einzelnen Produkte festgelegt und an die aktuelle Auslastung und die erwartete Nachfrage angepasst.

Für die Definition der zum Zeitpunkt t^k genutzten Instanz M^k des PO-Modells nutzen wir die zwei Entscheidungsvariablen-Familien y_p^k ($p \in P$) und $z_{p,s}^k$ ($p \in P, s \in S$), die die Kontingente für die einzelnen Produkte (y -Variablen) bzw. deren Aufteilung auf die verfügbaren Services (die z -Variablen) codieren.

$$\sum_{p \in P} REV(p) \cdot y_p^k \rightarrow \max \quad (1)$$

$$\sum_{p \in P} SPR(s(a), p, a) \cdot z_{p,s(a)}^k \leq C(a) \quad \forall a \in A \quad (2)$$

$$\sum_{s \in S} z_{p,s}^k = y_p^k \quad \forall p \in P \quad (3)$$

$$y_p^k \geq Y^k(p) \quad \forall p \in P \quad (4)$$

$$y_p^k \leq Y^k(p) + DTC^k(p) \quad \forall p \in P \quad (5)$$

$$y_p^k, z_{p,s}^k \geq 0 \quad \forall p \in P, s \in S \quad (6)$$

Die Summe der erwarteten Erlöse wird maximiert (1), in dem für jedes Produkt p die Anzahl y_p^k der anzunehmenden Aufträge bestimmt wird. Gallego et al. [1] schlagen die linearen Nebenbedingungen (2) - (6) zur Codierung der zulässigen Kontingente vor. Die Nebenbedingungen (2) limitieren die genutzten Ressourcenkapazitäten. Jeder angenommene Auftrag muss einem Service zugeordnet werden (3) und es wird sichergestellt, dass für jeden bereits angenommenen Auftrag ein passender Service verfügbar ist (4). Es können nicht mehr Aufträge akzeptiert werden, wie noch erwartet werden (5). Klein [2] führt aus, dass die explizite Berücksichtigung von Ganzzahligkeitsbedingungen für die Entscheidungsvariablen nicht notwendig ist.

Aus dem optimierten Tableau des Modells (1) - (6) können die Schattenpreise der Ressourcen $sp^k(a)$ aus der Zielfunktionszeile abgelesen werden. Sie stehen dort in der Spalte, die zur Schlupfvariablen der jeweiligen Kapazitätsrestriktion der Ressource a (2) gehört. Wir aktualisieren die Quoten $Q(p,s)$ durch die Zuweisung $Q(p,s) := z_{p,s}^k$ für alle Produkte $p \in P$ und alle Services $s \in S$. Schließlich werden die Bid-Prices durch die Zuweisung $BP(a) := sp^k(a)$ für alle $a \in A$ aktualisiert.

3 Umsetzung der Bid-Price-basierten Kapazitätssteuerung

Die Routine *capacity_control()*, deren Pseudocode in Bild 2 gezeigt wird, wird zur Bearbeitung des Online-Modells ($M^1, \dots, M^{N^{replan}}$) angerufen.

Zunächst wird das Netzwerk und der Buchungszeitraum spezifiziert (a). Anschließend werden die Systemzeit (b) und die Menge der akzeptierten Aufträge initialisiert (c). Falls die Buchungsperiode noch nicht beendet ist (d), so wird zunächst überprüft, ob ein Parameter-Update notwendig ist (e). In diesem Fall wird die aktuelle PO-Modell-Instanz erstellt (f), gelöst und die Bid-Prices aktualisiert (g). Anschließend wird auf eintreffende Aufträge gewartet (i). Sobald diese eintreffen, werden sie in *REQ* gespeichert (j) und die aktuelle Zeit festgehalten (k). Nun entscheidet das Request Acceptance Tool über die neu eingetroffenen Aufträge (l). Anschließend wird für die angenommenen Aufträge eine (vorübergehende) Ressourcen-Allokation durchgeführt (m). Die Annahme-Entscheidungen werden an die Kunden weitergegeben (n) und die Menge der angenommenen Aufträge entsprechend ergänzt (o). Anschließend wird der Parameter Cycle beendet (s). Falls die Buchungsperiode beendet ist, wird die Prozedur verlassen (q).

Die Funktion *process_by_bp()* realisiert das Request Acceptance Tool. Sie sortiert zunächst die neuen Aufträge (anhand eindeutiger ID-Werte) und entscheidet anschließend sukzessive über die wartenden Auftragsanfragen. Ein Auftrag wird angenommen, falls mindestens ein Service existiert, der ausreichend Kapazitäten bereitstellen kann und dessen Bid-Price geringer als der Erlös dieses Auftrags ist. Falls mehrere Services diese Bedingungen erfüllen, so wird der momentan behandelte Auftrag vorübergehend dem Service zugeteilt, der den geringsten Bid-Price unter den in Frage kommenden Kandidaten ausweist (Petrick et al., 2010a).

```

(a)  procedure capacity_control( $N, T^{max}$ );
(b)    current_time := 0;
(c)     $ACC := \emptyset$ ;
(d)    if ( current_time <  $T^{max}$  ) then
(e)      if (parameter_cycle_required) then
(f)         $M := \text{define\_parameter\_optimization\_model}(ACC)$ ;
(g)         $(BP) := \text{get\_control\_parameters}(M)$ ;
(h)      end;
(i)      wait for incoming requests();
(j)       $REQ := \text{fetch\_waiting\_requests}()$ ;
(k)      current_time := get\_current\_time();
(l)       $\text{process\_by\_bp}(REQ, N, BP, CAP)$ ;
(m)       $CAP := \text{make\_tentative\_assignment}(ACC, N)$ ;
(n)      propagate\_decisions();
(o)      update( $ACC$ );
(p)    Else
(q)      exit();
(r)    end;
(s)    goto (d);

```

Bild 2: Pseudocode des Kapazitätssteuerungs-Systems

4 Zuweisung von Aufträgen zu Fahrzeugen

Unmittelbar nach erfolgter Annahme wird für einen Auftrag ein ausführender Service ausgewählt. Eine mögliche Strategie besteht nun darin, diesen Service bis zum Ende der Buchungsphase beizubehalten (*NOFLEX*). Für Aufträge, die ein flexibles Produkt nachfragen, ist es aber möglich, die für die Auftragsausführung vorgesehenen Fahrzeuge (Services) einmal oder mehrmals zu ändern. Darüber entscheidet die Prozedur *make_tentative_assignment*(ACC, N), deren Funktionsweise im Folgenden beschrieben wird.

4.1 Zulässige Allokationen

Zum Zeitpunkt t^k muss jeder akzeptierte Auftrag einem der im Netzwerk N operierenden Services zugewiesen werden. Da die verfügbare Kapazität knapp ist, erfolgt diese Zuweisung modellbasiert durch Lösung eines ganzzahligen linearen Optimierungsmodells. Sei x_{rs}^k ($r \in ACC, s \in S$) eine Familie binärer Entscheidungsvariablen. Wir definieren $x_{rs}^k = 1$ genau dann wenn der Auftrag r zum Zeitpunkt t^k dem Service s zugewiesen wird. In einem Preprocessing-Schritt wird der Wert der binären Indikatoren $\rho(r, s)$ *derart festgelegt*, dass $\rho(r, s) = 1$ ist, genau dann wenn der Service s den Auftrag r ausführen kann, da er das Produkt $P(r)$ anbietet. Die Kapazität der Ressource $a \in A$ ist beschränkt durch $C(a)$.

$$\sum_{s \in S} x_{rs}^k = 1 \quad \forall r \in ACC \quad (7)$$

$$x_{rs}^k \leq \rho(r, s) \quad \forall r \in ACC, s \in S \quad (8)$$

$$\sum_{s \in S} \sum_{r \in ACC} SPR(s(a), P(r), a) \cdot x_{rs}^k \cdot cap(r, a) \leq C(a) \quad \forall a \in A \quad (9)$$

$$x_{rs}^k \in \{0, 1\} \quad \forall r \in ACC, s \in S \quad (10)$$

Jede Zuordnung von Aufträgen zu Fahrzeugen (Services) ist zulässig, solange die Restriktionen (7) – (10) eingehalten werden. Eine Partition der angenommenen Aufträge auf die vorhandenen Fahrzeuge wird durch die Nebenbedingung (7) sichergestellt. Solange (8) beachtet wird, ist sichergestellt, dass eine getroffene Zuordnung auch realisierbar ist. Die Restriktion (9) muss erfüllt sein, damit der Ressourcenbedarf der einem Fahrzeug zugewiesenen Aufträge nicht dessen maximal verfügbare Kapazität überschreitet. Schließlich können Aufträge nicht geteilt werden (10).

Nachfolgend beschreiben wir zwei Strategien zur Auswahl von Zuordnungen vor, bei der die Zuweisung von Aufträgen, die flexible Produkte nachfragen, in jedem Request Cycle geändert werden kann. Einerseits schlagen wir vor, eine möglichst geringe Kosten verursachende Zuordnung auszuwählen (*FLEX-BP*). Andererseits schlagen wir vor, nur dann Zuordnungen zu ändern, falls dies aus Kapazitätsgründen nicht verhindert werden kann (*FLEX-PA*).

4.2 Bid-Price-basierte Erst- und Neuzuordnung

Bei der Nutzung der Zuordnungsstrategie *FLEX-BP* werden zunächst die Kosten $BP(r, s)$ für die Zuordnung des Auftrags r zum Service s bestimmt und anschließend die gesamten Zuordnungskosten aller Aufträge minimiert (11). Die Kosten $BP(r, s)$ werden durch Summation der aktuellen Bid-Prices $BP(r)$ der bei Auswahl des Service s zur Erfüllung von r benötigten Ressourcen, festgelegt. Bei der Kostenminimierung werden die Nebenbedingungen (7) – (10) respektiert.

$$\sum_{r \in ACC} \sum_{s \in S} BP(r, s) \cdot x_{rs}^k \rightarrow \min \quad (11)$$

4.3 Re-Allokation bei minimaler Zuordnungsänderung

Die Motivation für die Definition der zweiten Zuordnungsstrategie *FLEX-PA* („Preserve Assignment“) besteht darin, einmal getroffene Zuordnungsentscheidungen möglichst beizubehalten. Für die Implementierung dieser Strategie wird die Zielfunktion (11) des Zuordnungsmodells durch die Zielfunktion (12) ersetzt. Der binäre Koeffizient $C(r, s)$ ist 1, falls Auftrags r zuletzt dem Service s zugeordnet wurde, ansonsten wird $C(r, s)$ auf „0“ gesetzt.

$$\sum_{r \in ACC} \sum_{s \in S} C(r, s) \cdot x_{rs}^k \rightarrow \max \quad (12)$$

5 Simulationsexperimente

In diesem Abschnitt wird über durchgeführte Simulationsexperimente unter Verwendung des vorgeschlagenen PO-Modells sowie der (Re-)Allokationsstrategien berichtet. Zunächst wird der Aufbau der Experimente skizziert (Unterabschnitt 5.1). Anschließend werden ausgewählte Ergebnisse präsentiert und interpretiert (Unterabschnitt 5.2).

5.1 Beschreibung der Experimente

Das vorgeschlagene Kapazitätssteuerungs-System wird für die Bearbeitung der in [5] definierten Auftragseingangsströme genutzt. Die Ankunft der Aufträge folgt einem nicht-homogenen Poissonprozess mit der höchsten Ankunftsintensität zur Hälfte der Buchungsperiode.

Fünf Fahrzeuge bieten insgesamt 5 Services an und verbinden dabei 18 Be- bzw. Entladeorte. Es werden 32 Ressourcen mit einer Kapazität von jeweils 50 Einheiten angeboten. Durch diese Services werden 106 Produkte, davon 32 flexible Produkte, angeboten. Jeder untersuchte Eingangsstrom umfasst 1250 sukzessive bekanntwerdende Aufträge. Wir betrachten drei unterschiedliche Auftragsströme $\omega \in \{1, 2, 3\}$.

Durch die Festlegung der (Re-)Allokationsstrategie $\Phi \in \{NOFLEX, FLEX-BP, FLEX-PA\}$ sowie der Parameter-Aktualisierungs-Zeitpunkte $N^{replan} \in N^* = \{10, 50, 100, 200, 500, 1000\}$ wird eine Konfiguration (Φ, N^{replan}) des Kapazitätssteuerungssystems definiert. Insgesamt werden $3 \cdot 6 = 18$ verschiedene Konfigurationen in $3 \cdot 18 = 54$ Simulationsexperimenten erprobt. Um einen Selektionsdruck zu simulieren, wird in den Experimenten die zukünftige Auftragsnachfrage um 10% unterschätzt.

Sei T^* die Menge aller Parameter-Update-Zeitpunkte. Für eine gegebene Konfiguration $r(t, \Phi, N^{replan})$ berechnen wir zunächst die bis zum Zeitpunkt t realisierte Summe der durch die angenommenen Aufträge erzielten Erlöse $r(t, \Phi, N^{replan})$. Sei nun r^{max} das Maximum der Menge $\{r(t, \Phi, N^{replan}) | t \in T^*, \Phi \in \{FLEX-BP, FLEX-PA\}, N^{replan} \in T^*\}$ und r^{min} das entsprechende Minimum in der Menge. Die skalierte Summe der bis zum Zeitpunkt t kumulierten Erlöse $r^*(t, \Phi, N^{replan})$ für die Konfiguration (Φ, N^{replan}) ist dann definiert durch $r^*(t, \Phi, N^{replan}) := (r(t, \Phi, N^{replan}) - r^{min}) / (r^{max} - r^{min})$. Diese Werte oszillieren zwischen 0 und 1.

Die Bid-Prices $BP(r, s)$ wurden auf der Basis einer Lösung des DLP (1) – (6) berechnet. Sie sind bei Beachtung der zugehörigen Kontingente $Q(r, s)$ optimal. Allerdings wird bei der Bid-Price-basierten Auftragsannahme nicht überprüft, ob diese Kontingente eingehalten werden. Daher ist es möglich, dass mehr Aufträge angenommen werden, als bei der Festlegung der Bid-Prices vorgesehen wurde. Wir vermuten, dass die Nichtberücksichtigung der mit der Lösung des DLP (1) – (6) verbundenen Kontingente möglicherweise zu einer Reduktion der insgesamt erzielten Erlöse führt. Daher beobachten wir den Anteil der Zuordnungen (r, s) unter allen tatsächlich durchgeführten Zuordnungen (r, s) , die über die optimalen Kontingente $Q(r, s)$ hinausgehen. Wir speichern diesen Wert in $o(t, \Phi, N^{replan})$. Anschließend werden die für eine Konfiguration beobachteten Werte in das

Intervall $[0;1]$ skaliert, in dem die skalierten Anteile der Quotenüberschreitung $o^*(t, \Phi, N^{\text{replan}})$ berechnet werden.

5.2 Ergebnisse

Die Anwendung von *NOFLEX* führt zu höheren Erlösen als die Anwendung der beiden Konfigurationen mit Neuzuweisung (Tabelle 1). Diese Beobachtung wird für alle Update-Häufigkeiten bestätigt. Die höchsten Erlöse werden erzielt, falls die Änderung von Allokationsentscheidungen verboten wird (*NOFLEX*) und nur wenige Parameter-Update-Zyklen durchgeführt werden. Aus diesen Beobachtungen schließen wir, dass eine Beibehaltung einmal getroffener Allokations-Entscheidungen vorteilhaft ist. Diese Schlussfolgerung wird durch die Beobachtung gestützt, dass die tendenziell zuordnungsbewahrende Strategie *FLEX-PA* zu deutlich besseren Ergebnissen als die *FLEX-BP*-Strategie führt und letztere Strategie einmal getroffene Zuordnungsentscheidungen überhaupt nicht zu bewahren versucht.

Φ	N^{replan}					
	10	50	100	200	500	1000
<i>NOFLEX</i>	1,00	0,90	0,89	0,86	0,93	0,95
<i>FLEX-BP</i>	0,28	0,03	0,00	0,21	0,42	0,29
<i>FLEX-PA</i>	0,48	0,10	0,31	0,23	0,49	0,33

Tabelle 1: Skalierte kumulierte Erträge $r^*(1.0, \Phi, N^{\text{replan}})$

Φ	N^{replan}					
	10	50	100	200	500	1000
<i>NOFLEX</i>	0,58	0,10	0,05	0,02	0,01	0,00
<i>FLEX-BP</i>	0,98	0,51	0,52	0,46	0,46	0,49
<i>FLEX-PA</i>	1,00	0,53	0,49	0,48	0,47	0,49

Tabelle 2: Skalierte Anteile der Kontingent-überschreitenden Zuordnungen $o^*(1.0, \Phi, N^{\text{replan}})$

Wir wollen herausfinden, warum die Modifikation von Allokations-Entscheidungen die Erlösmaximierung beeinträchtigt. Wir vermuten, dass die Bid-Price-basierte Auftragsannahme ohne Konsultation der zugehörigen Kontingente die Bid-Prices verfälscht und zu niedrig werden lässt. Daher werden Aufträge mit zu geringen Erlösen akzeptiert. Die für diese fälschlicherweise angenommenen Aufträge allokierten Ressourcen stehen dann für später eintreffende hochprofitable Aufträge nicht mehr zur Verfügung, so dass hier eine Erlös-Verdrängung stattfindet. Tabelle 2 stellt die beobachteten skalierten Anteile der Kontingent-Überschreitungen $o^*(1.0, \Phi, N^{\text{replan}})$ zusammen. Zunächst stellen wir fest, dass *NOFLEX* zu deutlich weniger Kontingent-Überschreitungen führt als *FLEX-BP* bzw. *FLEX-PA*, d.h. *NOFLEX* respektiert die Kontingente im größeren Umfang als die beiden anderen Strategien. Dies scheint ein erster Hinweis zu sein, dass die häufigere Überschreitung der Kontingente sich negativ auf die Summe der Erlöse auswirkt. Allerdings konnte bei einer Bid-Price-basierten Auftragsannahme mit Bewahrung der Kontingente keine signifikante Leistungssteigerung von *FLEX-BP*- und *FLEX-PA*-Konfigurationen im Vergleich zu *NOFLEX* beobachtet werden.

Um weitere Hinweise für die Gründe des unterschiedlichen Verhaltens der drei Strategien zu erhalten, haben wir die Entwicklung der Kontingentüberschreitungen $o(t, \Phi, N^{replan})$ während einer Buchungsperiode beobachtet. Bild 3 zeigt die beobachteten Werte für $o(t, \Phi, 10)$. Die Strategie *NOFLEX* hält die Überschreitungen der Quoten für die ersten 90% der Buchungsperiode stabil. Demgegenüber wachsen $o(t, FLEX-PA, 10)$ sowie $o(t, FLEX-BP, 10)$ kontinuierlich im Verlauf der Buchungsperiode. Bereits nachdem 40% der Buchungsperiode abgewickelt wurden, führen die beiden letztgenannten Strategien zu mehr Kontingentüberschreitungen als durch *NOFLEX* verursacht werden. Während bei *NOFLEX* maximal 4% aller Allokationen nicht mit den Kontingenten kompatibel sind, ist dies sowohl bei *FLEX-BP* als auch bei *FLEX-PA* in über 6% aller Allokationen der Fall.

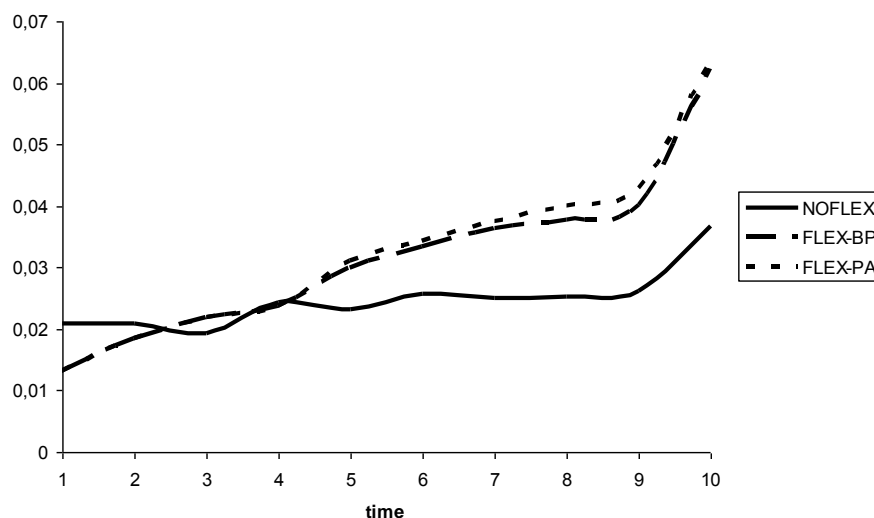


Bild 3: Anteil der Überschreitungen der Kontingente $o(t, \Phi, 10)$

Vergleichbare Ergebnisse haben wir in Experimenten mit einer sehr hohen Parameter-Aktualisierungs-Häufigkeit beobachtet (Bild 4). Bei der Anwendung von *NOFLEX* können keine Kontingentüberschreitungen festgestellt werden. Falls Änderungen der Allokations-Entscheidungen möglich sind (*FLEX-BP*, *FLEX-PA*), dann steigt die Häufigkeit der Kontingentüberschreitungen im Verlauf der Buchungsperiode auf bis zu 2.5% an. Überraschenderweise werden bei Anwendung von *FLEX-PA* mehr Überschreitungen festgestellt als bei der Verwendung von *FLEX-BP*. Die Erhöhung der Update-Häufigkeit und damit eine genauere Berechnung der Bid-Prices führen somit zu einer relativen Abnahme der Überschreitung von Kontingenten.

6 Zusammenfassung und Ausblick

Unsere Untersuchungen dynamischer Auftragsannahme-Situationen führten zu folgenden (Zwischen-)Ergebnissen: (i) Modifikationen einmal getroffener Zuweisungen von Aufträgen zu Fahrzeugen sollten vermieden werden, um Erlösverlusten vorzubeugen. Dieses Resultat bestätigt die einleitend formulierte Forschungshypothese H_1 . (ii) Anstelle von Änderungen der Zuweisung von Aufträgen zu Fahrzeugen ist es vorteilhafter, die Parameter der Annahmeregeln häufiger zu aktualisieren. (iii) Falls Änderungen der Zuordnung von Aufträgen zu Fahrzeugen (beispielsweise aus Kapazitätsgründen) nicht vollständig

vermieden werden können, so sollte die Anzahl der Modifikationen auf eine minimal mögliche Anzahl beschränkt werden. Diese Beobachtung bestätigt die eingangs formulierte Hypothese H_2 .

Als nächstes werden Untersuchungen über die Auswirkungen von Vorhersagefehlern durchgeführt. Des Weiteren werden die Einflüsse der Häufigkeit flexibler Produkte im Produktportfolio analysiert. Schließlich werden weitere Allokationsstrategien entwickelt und erprobt.

Danksagung: Dieser Forschungsbeitrag wurde durch die Deutsche Forschungsgemeinschaft im Rahmen des Sonderforschungsbereichs 637 "Selbststeuerung logistischer Prozesse – Ein Paradigmenwechsel und seine Grenzen" im Rahmen des Teilprojekts B7 unterstützt.

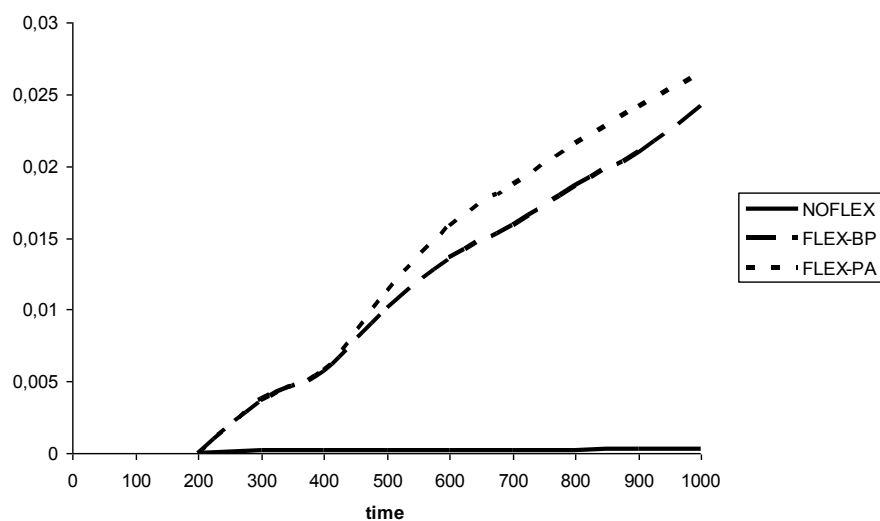


Bild 4: Anteil der Überschreitungen der Kontingente $\alpha(t, \Phi, 1000)$

7 Literatur

- [1] Gallego, G., Iyengar, G., Phillips, R., Dubey, A. (2004): Managing flexible products on a network. Technical report TR-2004-01, Columbia University, Ithaca, United States.
- [2] Klein, R. (2007): Network capacity control using self-adjusting bid-prices, OR Spectrum, Vol. 29, pp. 39-60.
- [3] Petrick, A., Steinhardt, C., Gönsch, J. and Klein, R. (2010): Using flexible products to cope with demand uncertainty in revenue management, OR Spectrum, DOI: 10.1007/s00291-009-0188-1
- [4] Rehkopf, S. (2006): Revenue Management-Konzepte zur Auftragsannahme bei kundenindividueller Produktion, DUV.
- [5] Schönberger, J., Kopfer, H. (2011): Approaching the application borders of network capacity control in road haulage, in Hülsmann, M., Scholz-Reiter, B. Windt, K. (Eds.), Limitations of Autonomous Cooperation and Control in Logistics, Springer-Verlag, Berlin Heidelberg

- [6] Schönberger, J., Kopfer, H. (2011): Revenue management in road-based freight transportation – impacts of uncertainty of capacity consumption, International Journal of Physical Distribution & Logistics Management, accepted for publication
- [7] Schönberger, J., Kopfer, H. (2009): Online decision making and automatic decision model adaptation. Computers & Operations Research, 36, 1740-1750.
- [8] Talluri, K. and van Ryzin, G.J. (2005): The theory and practice of revenue management, Springer.
- [9] Taha, M.A. (2007): Operations Research, 8th edition, Pearson Prentice Hall, Upper Saddle River.